John Delaney & Andres Fonseca
CMSC 12300 Project Report
June 4, 2013

Introduction

In short, we spent many weeks poking the data to see if our suspicions about the behavior of airline prices were founded in reality and then we used what we confirmed or denied to build a model that would predict whether the price of a flight would go up or down. It involved a lot of obstacles and false positives but we were eventually able to improve upon both random guessing. We struggled with strangely formatted data, the need to use three languages, and a real need to optimize.

At first we visualized the data and it's volatility by many categories, tried to reclassify the data intelligently, ran diagnostics on our assumptions, and then compared a couple models that used different explanatory variables. It forced us to become familiar with logistic regressions and what constitutes a good model. We built a web app that led us to believe that we could get extremely specific and make a model with hundreds of dummy variables that would do a good job. But, it taught us about rank deficiency and low resolution. We compared distributions across different categorizations of the data in order to infer sensible partitions within broader categories (i.e. could we do away with 7 dummy variables so that each day of the week had one or could we reduce it to 2 for weekdays and weekends). But, this taught us that many of our dummy variables were highly correlated and that doing better than random in a binary outcome is not very good.

Finally, we stepped back and tried to think of prices as a reflection of market demand and tried to get at the deviation from the price curve revenue managers at airline's try to adhere to and this proved to increase our predictive power.

Goals

We set out at the start of this project with three major interests regarding our data.

1. What do price paths of flights look like and what influences its shape? Working with data on flight fares gave us a lot of common sense ways of categorizing and visualizing the data but we wanted to get out the most meaningful or interesting among these.
2. Volatility. As econ majors, we wanted to use whatever familiarity with the data we gained to get the best handle on price behavior/misbehavior. We wanted to learn if there was a good way to trend or classify the data in a new or interesting way.
3. Prediction. Could we build a model that accurately predicts whether a flight's price will go up or down over an interval of time? This seemed like a simple enough task but things got hairy pretty quickly.

The Data

We were able to get our hands on a 60 files containing 20% of the daily flight information going in and out of O'Hare and Midway International Airport. Each file corresponds to a single query made around 2am every night where each row in the file is the information for a given flight returned by that query including information on fare price, departure date, airline etc. That is to say, every night a for two months, someone scraped that day's quoted price for American Airlines flight 153 a month from that night, a month and a day from that night, etc. This process was applied to 1200 unique flights each night in order to get the data we worked with. The files included all relevant information available to the clever company that got this data but we mostly used it to build a database that included:

- Departure date
- Airline
- Departure and arrival destinations (here on out we will refer to this as a 'market')
- Fare
- Time of day of departure

We figured these were the most relevant ways of categorizing the data. Furthermore, we knew that we wanted to categorize the data in other ways that we could calculate from this information (for example, the number of days until departure of a given quote).

Data Analysis Techniques And Technology Used

Besides getting our hands dirty exploring the data's structure, shape, and categories, we used some models for prediction that we hadn't covered in class.

More specifically, we knew that implementing a logistic regression would be a sensible way to predict a binary outcome (did the price go up or down).
In statistics, logistic regression is a type of regression analysis used for predicting the outcome of a categorical dependent variable (a dependent variable that can take on a limited number of values, whose magnitudes are not meaningful but whose ordering of magnitudes may or may not be meaningful) based on one or more predictor variables. We ended up using the following as predictor variables:

- Normalized fair (percent deviation from the average)
- Trailing percent price difference for 'i' periods (i.e. the percent price difference looking back 'i' days.
- Dummy variables for airline, market, time of day, day of the week the flight leaves

Thus, our implementation involved building a matrix (in the form a of a 'data frame') by massaging and reshaping the data to include the aforementioned variables.

With that in mind we also used many new libraries and tools. In part because we technically used R, sqlite3, and python to accomplish our tasks but also because of our focus on optimization and trying to use each language in only the most efficient ways we could. So, we tried to keep the bulk of our code within python.

The following is a list of the more important libraries we used in order to accomplish our tasks:

- Pandas: it allowed us to get R-like functionality with data frames in python. This went a long way to optimizing our code
- Pyper: actually allowed us to call R from python to use R's built in logistic regression and error calculation tools
- Sqlite3: allowed us to build our database from which we could build the aforementioned data frame. It was helpful to have an updated and saved copy we could query for the sake of testing instead of scraping the ftp every time we wanted to build a new data frame that included new columns.
- Numpy/itertools: both played a large part in making our code run quick enough.

In fact, numpy and itertools deserve special attention because our previous implementations were extremely costly before we optimized. In fact, as we'll discuss soon enough, we started down a journey of comparing different models in order to see if we could improve our predictive power and this optimization is essentially the only way we could do it.

Implementation

We will suppress any conversation about previous implementations of our code for now. This project's history, though suspenseful and intricate, will serve more to describe what we learned from failure rather than the architecture of our final results.

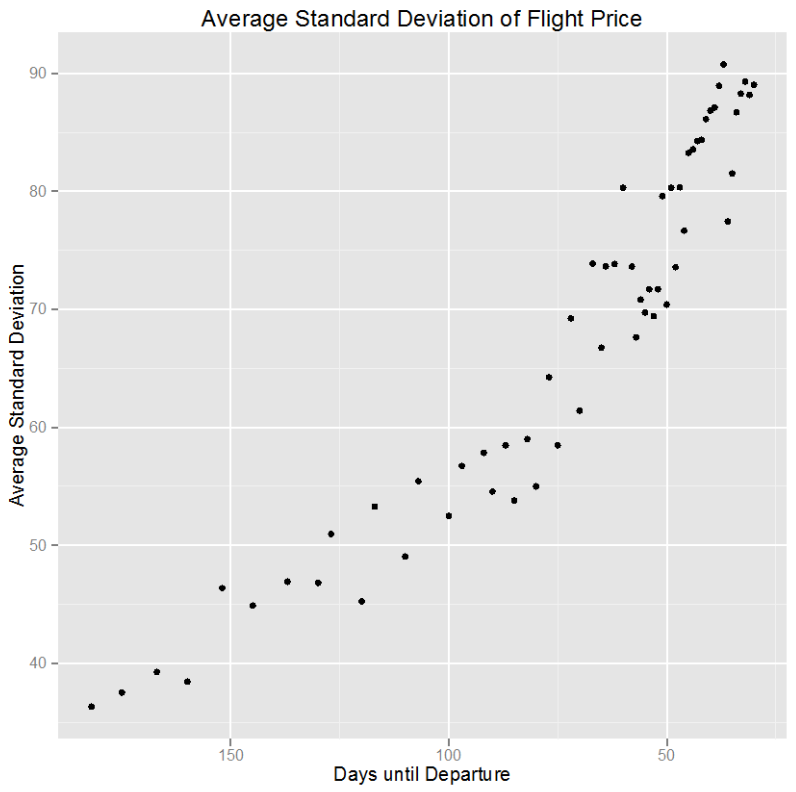
Our implementation is divided into the following (not necessarily discrete) categories:

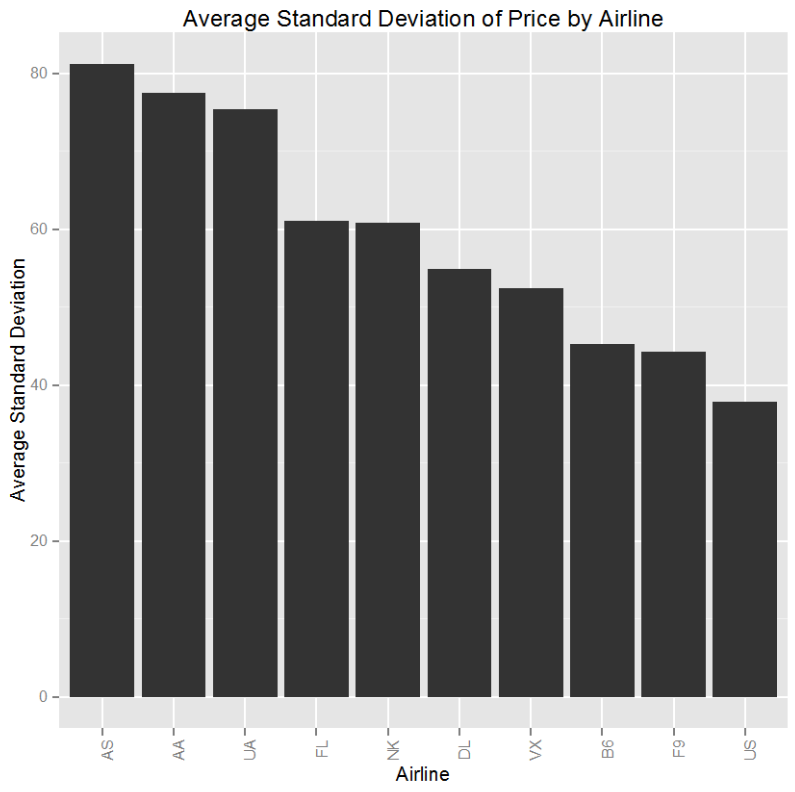
- Data exploration
- Data analysis
- Data retrieval/database construction
- Results, visualization, and testing

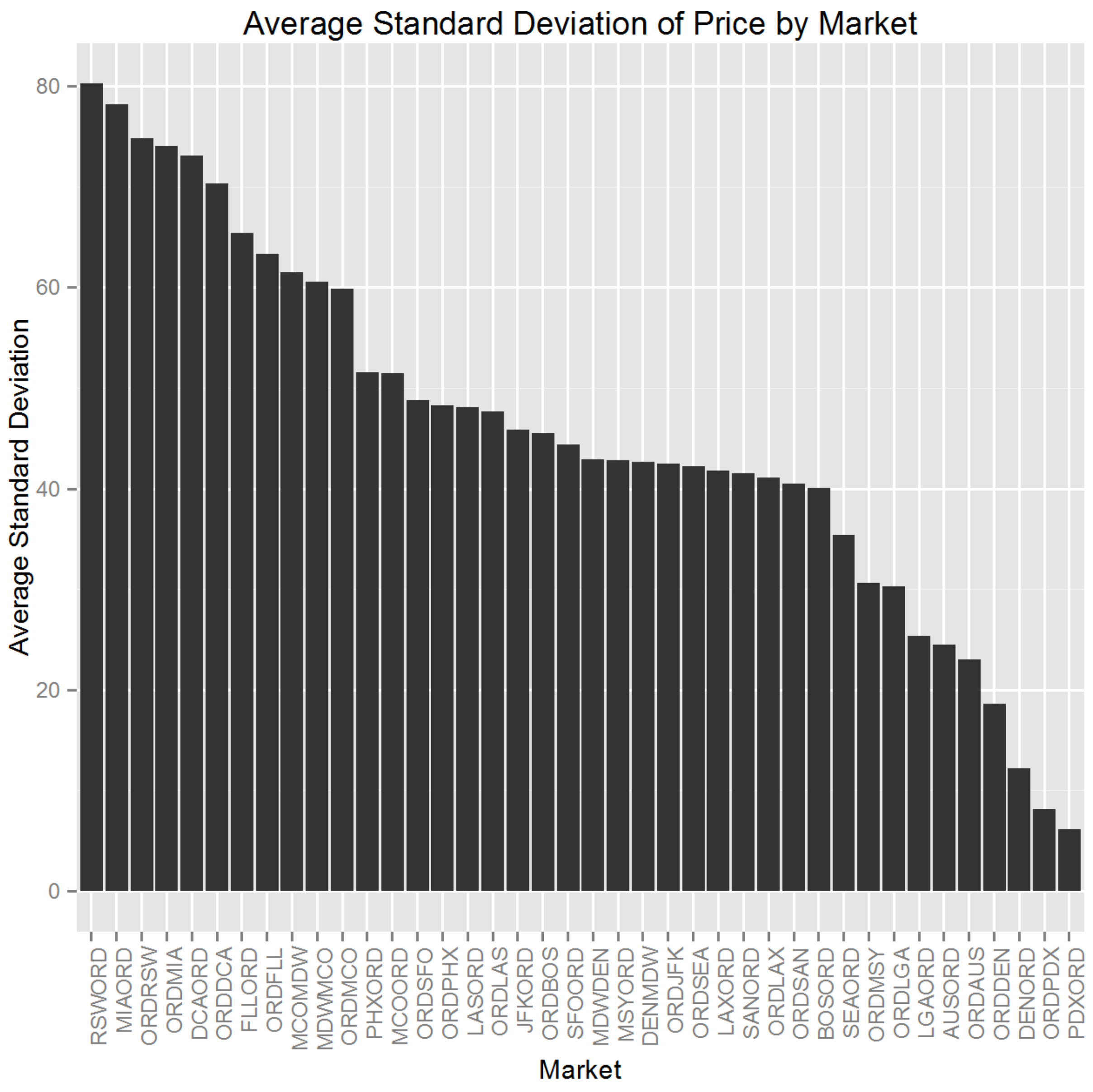
On the whole, our code reflects the conversation between us and the data. A thought or question would occur to us, it would fit into one or more of these categories and we'd do some research into the best way to satisfy our curiosity.

In the beginning we needed a way of getting our data and putting it somewhere we could manipulate. In short, we built a database. Our data was hosted on an ftp server and a new file was getting added every night so we had to write some code that would access the ftp, check for new files periodically, and update our database. We used python to check the ftp and download files and sqlite3 to build/update our database.

The next step involved a lot of exploration, visualization, and R. Our data was unorganized and unsorted. There were simple descriptive statistics of samples of our data that would teach us a lot but there was a smaller scale of analysis and visualization involving the history of a flight's price path that was important to us as well. For example, given our interest in volatility, we could analyze the data by category like so:


Average Standard Deviation of Flight Price
Average Standard Deviation
Days until Departure
90
80
70
60
50
40
150
100
50


Average Standard Deviation of Price by Airline
Average Standard Deviation
Airline
80
60
40
20
0
AS
AA
UA
FL
NK
DL
VX
B6
F9
US


Average Standard Deviation of Price by Market
Average Standard Deviation
Market
80
60
40
20
0
RSWORD
MIAORD
ORDRSW
ORDMIA
DCAORD
ORDDCA
FLLORD
ORDFLL
MCOMDW
MDWMCO
ORDMCO
PHXORD
MCOORD
ORDSFO
ORDPHX
LASORD
ORDLAS
JFKORD
ORDBOS
SFOORD
MDWDEN
MSYORD
DENMDW
ORDJFK
ORDSEA
LAXORD
SANORD
ORDLAX
ORDSAN
BOSORD
SEAORD
ORDMSY
ORDLGA
LGAORD
AUSORD
ORDAUS
ORDDEN
DENORD
ORDPDX
PDXORD

The first figure represents an important conclusion in that it confirmed that which we were assuming about fare volatility as we approach the date of departure. The second figure confirmed our suspicion that not all airlines have the same relationship to volatility but that all nine accounted for in our data do not differ drastically and could likely be described on similar scales. The third figure confirmed a less obvious thought that some flights probably behaved (at least with regards to volatility) in similar ways. This thought is especially affirmed by the fact that all departure-arrival pairs contain their return trip within the same chunk if not next to each other on the sorted list. In order to implement this exploration we used R, ggplot2, and sqlite3.

While this helped confirm or deny suspicions about the data before we wasted time constructing models, we needed to see how the price of a historical flight evolved as it approached its departure date. We needed to see these paths and even compare paths we considered similar to each other. We stayed within R but decided to build a locally hosted website with a package called Shiny that would allow us to use an interactive GUI to display these paths.

But, the bulk of our implementation involved getting our logistic regression working. As we discussed, we took files from an ftp, updated a database, calculated several new columns that we would use as explanatory variables such as:

- What day of the week did we get the quote
- What day of the week was the departure date
- When did the flight leave during the day? Morning? Afternoon? Evening? Red-eye?
- Days until departure
- The fare's deviation from the average for that market that many days out

From here the main problem faced in implementing our code was dealing with memory constraints on our local machines. Our initial implementation in R involved running a logistic regression with dummy variables for airline, departure day of the week, time of day (morning, afternoon, evening, etc.) and then considering all of the cross terms. The result of this is that the size of our dataset in memory began to grow well beyond what was feasible for our machines because we were adding hundreds of columns to a data set that was already had hundreds of thousands of rows. Our solution to this was to rebuild our solution from the ground using python with a focus on using subsets of the data rather than perform analysis in one fell swoop.

The architecture of our final solution consists of two main files, one to retrieve data from the ftp and store the results in a local database, and the other to read data from the local database for analysis. The first file operates by loading the list of files on the ftp, checking that list against a local list of parsed files and then downloads each file at a time to a local memory, loading each into the database as it goes. The second file reads data from the local database, preprocesses that data and then calls out to r using pyper to perform our analysis. The pandas library does the majority of the heavy lifting in this section because it allows us to mimic the functionality of R data frames with numpy like performance.

Results

We set out to understand the behavior of flight fares going to and from Chicago airports with the hopes that we could build a model that could guess whether the price would go up or down over an interval of time (as an input). We spent a lot of the process exploring and visualizing our data in order to prevent ourselves from going to down a fruitless path. For many reasons we were unsuccessful in preventing this. We entertained many ideas about how to classify the data in a more meaningful way than by the natural categories presented to us in the construction of our database. For instance, we tried using a metric that compared distributions—at first with fares (in order to find a robust way of categorizing different fare structures within a flight's published quotes) and then by days to departure in order to decrease the number of dummy variables in our regression—but the resulting heat maps and partitions only revealed strong correlations among different factors we were considering.

Our dataset of over 500 MB was large enough for us to worry about optimizing our code in every way possible but small enough to run on our personal machines. So we tried to simplify the problem the best we could and optimize our code accordingly. We hoped this would amount to a model we could cross validate and test often. Eventually, we did get such a model that reported a 23.5% error, which we were proud of and encouraged by. Unfortunately, when we counted all the recorded increases and decreases in fare that our model used to build the regression, we found that the price went up nearly 75% of the time. Which means that a model that guessed that the price would go up (over the next three days, in this case) every single time would only do about 2% worse than our logistic regression.

To come to our final analysis we rethought our approach to modeling the system. Instead of throwing in all of our variables and hoping for the best, we built up a series of assumptions about what could increase the predictive power of our model. First, we realized that our analysis rests on the assumption that fare price is a reflection of consumer demand for a given flight. Thus, if we wish to predict changes in fare price, we must realize that we are actually trying to predict changes in consumer demand. Given that there are only a finite number of seats on a plane, as seats are bought up on the plane, supply is constricted so demand increases. Thus we are faced with the problem of prices that tend to increase over time on average. In addition, given that a price a reflection of demand, we care more about the relative price than we do the absolute price. A $500 fare from Chicago to New York reflects a "willingness to pay" ie demand that is much higher than a $500 fare from Chicago to London. Thus, we wanted to structure our analysis so we considered each flight price within a context of similar flights. We defined similar flights as flights that are each x days from departure that leave on y day of the week that leave in the jth part of the day. Using this set of similar flights we calculated the average for this set and then calculated the percent deviation from this average for every point in the set. Our intent with this transformation was included information about other flights into the regression for a given flight. That is it takes into account competition by making every flight a reflection of it's deviation from the price you would expect to see. In addition to establishing a notion of "high" prices and "low" prices through this normalization process we also wanted to take into account the past movements of a price path into consideration. This was achieved by calculating the trailing change in price back one period and two periods. By adding a normalized price and trailing prices changes to the regression we hoped to add predictive power to our model given our assumptions. In practice however, we see gains that are marginal at best. When considering the route from O'Hare to San Francisco we see less no real increase in predictive power by using normalized fare over raw fare and with controls we actually see a decrease in predictive power when we add the lagged changes in prices to our regression. If we consider the route from O'Hare to Los Angeles we see similar results for the difference between the raw fare and normalized fare but now an increase in predictive performance with adding the lagged terms. We see similar results across the dataset indicating that using normalized fare has no effect on predictive power and using lagged prices has mixed results. The only consistent result is that we are able to achieve marginal improvements over using random data by using these factors. The conclusion we can draw from this is that these variables do have some degree of predictive power. The problem is that they are all effectively proxies for what we actually want to measure, demand. In order to increase our predictive power we need to consider other variables that are better proxies for demand. I would argue that it would probably be much easier to predict fare price using kayak search histories than it would be to use just fare prices alone.